WiseType: A Tablet Keyboard with Color-Coded Visualization and Various Editing Options for Error Correction

Ohoud Alharbi[1], Ahmed Sabbir Arif[2], Wolfgang Stuerzlinger[3], Mark D. Dunlop[4], Andreas Komninos[5]

[1,3]Simon Fraser University, [2]University of California Merced,[4]University of Strathclyde, [5]University of Patras

ABSTRACT

To address the problem of improving text entry accuracy in mobile devices, we present a new tablet keyboard that offers both immediate and delayed feedback on language quality through auto-correction, prediction, and grammar checking. We combine different visual representations for grammar and spelling errors, accepted predictions, and auto-corrections. and also support interactive swiping/tapping features and improved interaction with previous errors, predictions, and auto-corrections. Additionally, we added smart error correction features to the system to decrease the overhead of correcting errors and to decrease the number of operations. We designed our new input method with an iterative user-centered approach through multiple pilots. We conducted a lab-based study with a refined experimental methodology and found that *WiseType* outperforms a standard keyboard in terms of text entry speed and error rate. The study shows that color-coded text background highlighting and underlining of potential mistakes in combination with fast correction methods can improve both writing speed and accuracy.

Keywords: Text entry; predictive text; virtual keyboard; touchscreen; auto-correction; error correction; error detection; backspace.

CCS Concepts: Human-centered computing → Text input; Human-centered computing → Touchscreens; Applied computing → Text editing

1 INTRODUCTION

Today, the use of touchscreen text entry extends from short text messages to longer emails and blog posts, where the latter categories require more formal writing and text entry errors are less tolerated. Yet, many touchscreen users encounter problems during text entry, especially when entering longer text passages. One of the challenges is the limited speed of text entry on touchscreens, which is substantially slower than physical keyboards and well below the inviscid text entry rate where the keyboard ceases to be a barrier to communication speed [28]. An even more notable challenge is the reduction of writing errors during text entry, which can be frustrating with correction considerably compromising entry (e.g., [4]). Mobile device manufacturers address spelling errors by providing auto-correction and adding various forms of predictive text entry. However, auto-correction can lead to confusing or embarrassing inadvertent mistakes and many users turn off auto-correction features [6,33], presumably because they perceive mis-corrections as overly annoying. Our research attempts to ameliorate these issues through a virtual keyboard that includes both a spelling and a grammar checker, provides augmented visualization for previous automatically or manually corrected errors, and enables smart swipe and tap features to speed up the error correction process. Our main goals are a) to better inform users about automatic changes to the inputted text that may occur during text entry and b) to decrease the overhead of correcting errors without negatively affecting typing speed.

In our new input method, we extend the concept of highlighting spelling mistakes to increase user awareness, as shown in [26], through color coding for spelling and grammar mistakes. We show errors with attention-grabbing colors, such as red and orange, for spelling and grave grammar mistakes, respectively, while more neutral colors, such as blue and gray, are used for auto-correction results and accepted predictions. This new input method is designed to help users to spot mistakes, identify their cause easily, and then enable them to quickly go back and edit the text, without slowing down the text entry speed.

After the discussion of related work, we present the results of four studies: three pilot studies (N1 = 7, N2 = 9, N3 = 9) to refine feedback and experiment design, and a main study (N = 12) on the effect of our novel approaches on text entry behavior.

2 RELATED WORK

Text entry assistance like auto-correction is important for fast text entry and efficient communication between text message users [10]. Such assistance is beneficial when it works as expected. However, it can also cause significant communication problems when it fails [10]. Moreover, the cost of error correction can also be high [5].

Error correction has been shown to be particularly important on touchscreens with small keys (e.g., [29]) and was seen as one of the challenges for intelligent text entry [45]. However, auto-correction errors, such as failures of current correction mechanisms, are widely discussed in the press as a problem of modern mobile text entry (e.g., [49]) and can lead to unintended messages being sent.

Support for error awareness and correction have been identified as a strongly desired feature in smartphone keyboard design studies [25], especially for older adults (e.g. [26]) and children (e.g. [7]).

There are several studies that introduce innovative text entry methods, such as key-target resizing keyboards (e.g., [21]), gesture keyboards (e.g., [1,11,34,40,52]), tap-stroke hybrid keyboards (e.g., [3]), keyboards with alterative layouts (e.g., [12,16,51], and keyboards that adapt to the hand posture (e.g., [9,19]). However, the focus is usually either on novel devices/techniques or on speed – but not on error correction.

Predictive systems can support effective error correction and completion of partially entered words if an appropriate language

[1] ohoud_alharbi@sfu.ca, Computing Science @ Simon Fraser University, CA.
[2] asarif@ucmerced.edu, University of California-Merced, USA.
[3] w_s@sfu.ca, The School of Interactive Art and Technology (SIAT) @ Simon Fraser University, CA.
[4] mark@textentry.org.uk, University of Strathclyde, Glasgow, UK.
[5] akomninos@ceid.upatras.gr, University of Patras, GR.

model is used [18]. Some systems also provide phrase prediction while writing (e.g., [8]). Word prediction and showing keystrokes could save up to 45% keystrokes in mobile keyboards [18], but this promise is rarely transferred to a corresponding increase in typing speed due to the higher cognitive load for handling word completions [15,24,39]. Other systems, such as the Smart-Restorable Backspace [2], suggest text that was previously written or deleted by the user to help them recover faster from mistakes.

2.1 Text Entry Evaluation

Allowing participants to type whatever they want in user studies may seem to be a desirable approach, as it replicates natural user behavior. However, this confounds experimentation, as there is a lack of control for performance measurements [32]. Accuracy is difficult to calculate since there is no source to compare the written text with [32] and users may "game" the experiment by entering gibberish or many short, easy words. Hence, although alternative text entry tasks have been proposed (e.g., [35,49]), the standard procedure in lab-based test entry studies is to show participants pre-selected phrases one at a time and ask them to transcribe them [40].

To increase comparability between studies, standard phrase sets are often used. As the primary metric of text entry studies is input speed, most studies use phrases that are relatively short, easy to remember, and representative of the target language. The most common phrase set in studies is from MacKenzie and Soukoreff [32]. One issue with these phrases is that they have not been written by mobile device users and do not look like actual mobile messages [48]. The MobileEmail collection [48] contains emails written on mobile devices. Of particular importance is the memorable phrase sub-set that has been tested for memorability in transcription tasks. To simplify the study procedures and to eliminate any potential confounds, phrase sets usually do not contain punctuation symbols, and require a few or no uppercase letters. Further, participants are often instructed to ignore uppercase letters and enter all text using lowercase. Mobile phrase sets also have higher proportions of sentences in the first person and sentences that are questions [48].

In the literature, there are three approaches to error correction conditions in text entry user studies: no error correction required, recommended, and forced error correction [4]. In the first condition, participants are not required to correct errors. In the second, participants are recommended to correct errors as they spot them. With the third condition, participants are forced to correct all errors in order to proceed to the next phrase. Typically, error rates are very low (e.g., [28]) in laboratory text entry studies, even on very small screens (e.g., [20]), as users tend to pay close attention and trade off speed for correctness. In our work, we used the second condition, where we recommended to participants that they correct errors, since we wanted to compare the text accuracy with our auto-correction method augmented with visual feedback and typical auto-correction without feedback.

3 Motivation

As discussed above, errors are costly in time, effort, and user perception of text entry quality, yet autocorrect is only a partial solution. Recent research suggests that increasing the visibility of suggestions can increase both perception and interaction costs, which could reduce text entry speed (e.g., [23,30,38,39]) and in some cases seems to eradicate all benefits in writing accuracy [8].

Offering multiple choices for corrections can give users more freedom. This could provide notable benefits, because it gives users the flexibility to deal with a wide range of needs and situations and also helps users who benefit from good modality choices [37]. In other cases, offering multiple choices for corrections might lead to less favorable results with users who make bad choices, who then also prefer systems that offer limited interaction possibilities [23].

Our goal is thus to find a reasonable combination of methods that helps users identify and manage errors, while still increasing average typing speed. We formulated the following research question to drive our work in this paper: *Does the combination of more assertive error visualizations with fast correction methods improve writing speed and accuracy in text entry?*

4 Exploratory Studies

Our design goal was to create an interface that avoids overloading users visually, while at the same time making it easier to spot grammatical and spelling errors, and (potentially incorrect) auto-corrections. We used an iterative design process to carefully evaluate our design choices for the keyboard in pilot studies to pick the most effective features. We then evaluated the final prototype in a full user study. The exploratory studies were:

1. Study of initial error visualizations;
2. Study of revised error visualizations;
3. Pilot comparison of *WiseType* with standard keyboard.

Following a previously used data collection approach [27]. we used a set of phrases randomly chosen from a set of user reviews of local venues, submitted to FourSquare[6], written in English, in these pilots. We analyzed phrases using the LanguageTool.org grammar checking API with the following rules enabled: Collocations, Commonly Confused Words, Core Grammar, Nonstandard Phrases, and Possible Typo. We then manually scanned phrases to exclude those where either the errors were minor or arguable, or where there were undetected errors. We then further filtered the set to arrive at two collections of phrases: clean, with zero errors, and problematic, with one or two grammatical errors per phrase.

4.1 First Pilot Study: Highlighting

The aim of this study was to collect user opinions about the first iteration of *WiseType*, which highlighted all mistakes, auto-corrections, and predicted words in different colors (see Figure 1). Here, warm colors (red and orange) are used to warn about serious and minor writing errors, while cool colors (shades of blue) reflect system-generated corrections, which might be potentially incorrect.

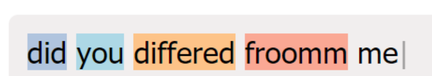

Figure 1: The visual feedback used in the first pilot study.

We recruited 7 participants (6 female) for this pilot. Each participant entered 10 phrases from the above set, all of which contained grammatically incorrect English. We suggested that participants correct the writing errors, but we did not force correction. At the end of this session, we conducted a semi-structured interview to ask them about the proposed visualizations and elicited additional suggestions. The whole session lasted 30

[6] https://foursquare.com

minutes. The results from interviews and observations revealed that the participants spent more time transcribing the phrases than usual, because they perceived highlighted auto-corrected and predicted words as mistakes, not as corrections. Some users even forgot what each color represented. This encouraged us to adopt a different form of visualization to better distinguish for potentially incorrect auto-corrections than for other errors.

4.2 Second Pilot Study: Highlighting and Underlining

In this study, we aimed to validate the re-design with different forms of visual feedback for errors. We compared two approaches: underline only vs. the combination of highlight and underline (Figure 2). We recruited 9 participants for this pilot (8 female). Six participants reported that English is their second language. We counterbalanced the order of conditions. After 2 training phrases, each participant entered 10 phrases per condition and completed a pre- and post-session questionnaire. At the end, we conducted a semi-structured interview to ask about their perception of the design and their suggestions. Sessions lasted 30 minutes. The questionnaire and interviews showed that participants preferred the combined highlight and underline approach (Figure 2 bottom) and also remembered the meaning of each color well.

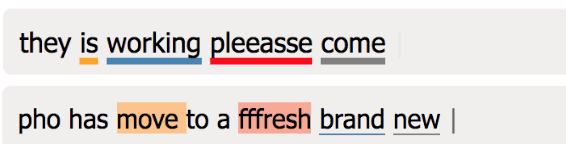

Figure 2: The visual feedback used in the second pilot study.

4.3 Third Pilot Study: Validation

This pilot compared *WiseType* with a conventional keyboard. We recruited 9 participants (8 female). All participants had English as their second language and their mean IELTS score was 6.33/9. We used a within-subject design with two keyboard conditions (conventional and *WiseType*). We counterbalanced the order of conditions. After 2 training phrases, each participant entered 10 phrases per condition and completed a pre- and post-session questionnaire. At the end, we conducted a semi-structured interview to ask about their perception of the design and their suggestions. Each session lasted 30 minutes.

The conventional keyboard has the same buttons dimensions and response time as *WiseType*, and works like most touchscreen keyboards with auto correction enabled (but uses the same algorithm as *WiseType*), has a prediction panel with 3 options, and represents spelling mistakes through a red underline. Yet, it does not visualize errors.

We measured the error rate (see section 7.3 for details) of the submitted phrases from our logging data. The mean number of writing errors with the conventional keyboard (M = 8.70, SD = 5.85) was higher than with *WiseType* (M = 4.88, SD = 3.00) and this difference is statistically significant, $F(1,26) = 16.784$, $p = .000$, with medium to large effect size (Cohen's dz) = 0.79 and study power $(1-\beta) = 0.70$. The data shows that writing errors decreased when using *WiseType* (see Figure 3).

In the post-session questionnaire participants responded that they would use the new keyboard on their touchscreen devices, and they rated the overall experience with the keyboard positively: 7 (77.78%) answered either "*excellent*", "*very good*", or "*good*".

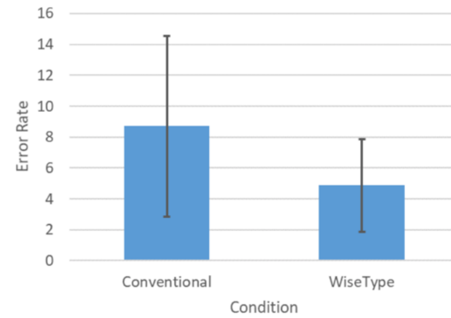

Figure 3: Average Error Rate for both text entry conditions (y-axis represents percentages).

5 THE *WISETYPE* KEYBOARD

The pilot described above identified that our approach can improve writing accuracy, but our work aims to improve writing speed as well. Thus, we added a feature which allows users to tap on highlighted/underlined words to correct them via a context menu and measured the typing speed in the main study. We also noticed that the phrase set used in the pilot contained slang and unfamiliar words (where participants even asked the observer for meanings), which created confusion and reduced entry speed. The Singapore SMS Corpus [14] has been similarly criticized for "strange language, abbreviations, and sentence fragments", leading to confusion and high error rates [48]. Thus, we decided to change to the Enron MobileEmail phrase set [48] into which we selectively injected various errors (discussed below).

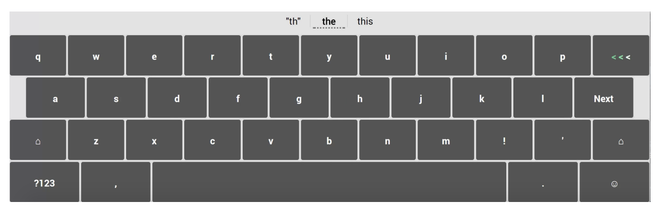

Figure 4: *WiseType* Keyboard.

WiseType was implemented as a web application using JavaScript and PHP. As stated above, we iterated on the design through multiple pilots. In the following paragraphs, we outline the final design and the inclusion of better mechanisms to support error detection and correction.

5.1 Word Prediction and Auto-Correction

WiseType includes standard predictive keyboard features, similar to the default Android text entry method. It displays a prediction panel above the virtual keyboard which show three candidate suggestions: the original text on the left, the most probable prediction in the center (highlighted with an underline), and the second most probable one on the right. As with other systems, *WiseType* allows users to choose the candidate word by pressing the space key or by tapping on the candidate word in the panel (see Figure 4).

WiseType also triggers auto-correction for spelling mistakes and replaces a misspelled word with the most probable correct word. The auto-correction is triggered when the Levenshtein distance [31] between the inputted and the candidate word is a single operation.

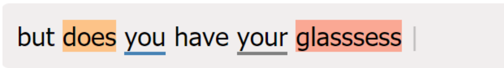

Figure 5: The visual feedback for the final *WiseType* design.

5.2 Highlight and Underline

Misspelled words are highlighted using colors that reflect the severity of the mistake: red for spelling mistakes and orange for grammar errors. Blue underlining is used to show instances of auto-correction and gray underlining denotes instances where the user picked a word from the prediction panel (see Figure 5). The motivation for showing words that the users picked from the prediction panel is that the user might make a mistake while picking a prediction. With the gray underline the user can quickly see where he/she accepted predictions and then later go back and fix any mistake, if necessary, without having to re-read the whole text.

Recently, some smartphone manufacturers added a context menu option to enable users to edit auto-correction mistakes[7]. Yet, this functionality is very limited as it only provides a *single* prediction and an option to undo auto-correction. *WiseType* instead offers up to six predictions, one of which un-does the auto-correction. Furthermore, *WiseType* makes it visually easier to identify (potential) errors (see Figure 6). As with other touchscreen keyboards, a tap and hold gesture on a character moves the cursor, while a simple tap on a highlighted/underlined word activates the context menu, as in Figure 6.

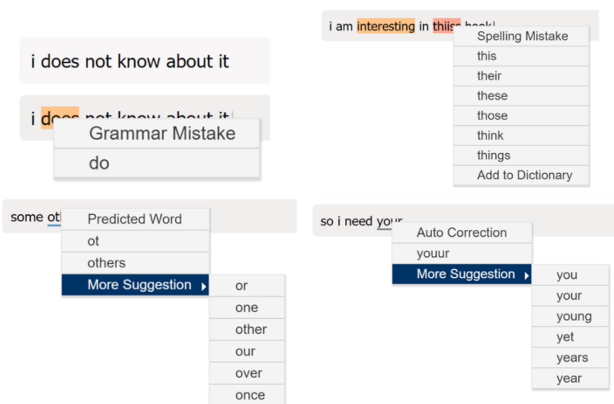

Figure 6: The *WiseType* context menu for a grammar mistake is shown in the top left, the spelling mistake correction menus (similar to standard mobile spell correction) in the top right, our novel predicted word expansion in the bottom right, and the auto-correct adjustment menu in the bottom left.

For misspelled words, *WiseType* allows users to add a non-dictionary word into the dictionary by tapping on the word and choosing the "*Add to dictionary*" option from the context menu. Once a word has been added to the dictionary, the system does not consider it as a misspelled word in the future. To guard against errors in this operation, *WiseType* enables users to delete a mistakenly added word from the dictionary by tapping on a word and selecting the context menu entry that removes the word from the dictionary.

5.3 Super Swipe

Smart deletion is automatically triggered when users perform a swipe-left gesture on the backspace key. *WiseType* will then delete the text right up to the last word with a grammar or spelling mistake. In contrast to previous work, the swipe technique in *WiseType* is applies to multiple errors: grammar and spelling mistakes.

However, we noticed during the pilot that participants did not use this feature much. When queried as to why they had not used it, several said that they had forgotten about the feature. One participant recommended using a different icon to represent the swipe feature. Based on this idea, we changed the appearance of the backspace icon from the default one (see Figure 7). After deleting a chunk of text through the smart deletion feature, *WiseType* suggests text from the deleted chunk in the prediction panel.

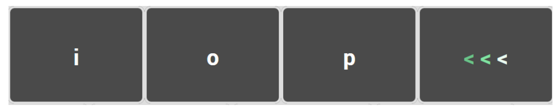

Figure 7: The super swipe button.

5.4 Grammar Checker

We used a cloud-based grammar checker, LanguageTool, which provides a JSON API[8]. LanguageTool's response to input is an array of possible grammatical errors, each with an explanation and suggested correction. Since the phrases we used had no punctuation and were all in lowercase [32], we filtered all responses during the study to remove any errors related to punctuation and capitalization mistakes. Unlike feedback for spelling mistakes, which appears immediately, the grammar checker is invoked only after typing at least three words, since many grammatical errors need more context before being detectable. The two-word entry *"The girls"*, for example, cannot be checked grammatically until more text is written such as *"…are over there"* (correct) or *"…was over there"* (incorrect). As in this example, grammatical errors are thus often identified in earlier parts of a sentence relative to the current typing position, which causes them to be outside of the focus of attention for (at least some) users. This is a similar situation to some of the most powerful auto-correction algorithms, such as VelociTap [50], which perform best with the whole sentence to analyze – which again results in similar post-hoc corrections far from the typing position. *WiseType* deals with this issue by highlighting the grammatical errors with more attention-grabbing color (orange).

6 Main Experiment

The purpose of this study was to compare *WiseType* with a conventional predictive keyboard. Both keyboards are implemented in JavaScript using the same predictive and autocorrection algorithms, and the of the conventional keyboard has the same button dimensions and response time as *WiseType*. The conventional condition looks like most touchscreen keyboards, has auto correction, yet without any feedback when it changed a word, does not show grammar errors, has a prediction panel with 3 options, and represent spelling mistakes with red underline.

6.1 Apparatus

We used a Microsoft Surface Pro 3, Intel Core i5 processor, 29.21×20.14×0.91 cm, and 798 grams, with Windows 10 at

---

[7] https://www.samsung.com

[8] https://languagetool.org/api/v2/check

2160×1440 resolution. The key dimensions were 80*90 pixels (21.2*23.8 mm). Figure 8 illustrates the physical setup.

We used a tablet instead of smartphone because tablets are equipped with larger screens allowing easier viewing and typing. Tablets are typically used for activities that are centered around heavier text composition (e.g., email, social media) [47] and are often used in professional contexts where text entry accuracy can be critical (e.g., medical settings) [17,42,46]. Furthermore, using tablets helps isolate problems emerging from "fat-finger" concerns [43]. We specifically used a high-powered tablet to ensure that processing delays were not a limiting factor for our work.

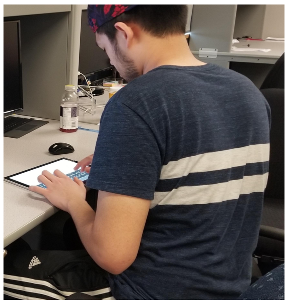

Figure 8: The physical setup used during the experiment.

6.2 Design

We used a within-subjects design with two conditions. In each condition, participants entered 23 phrases (46 phrases in total, excluding practice phrases). Hence, the design was:

12 participants ×
2 conditions (conventional and WiseType) ×
23 phrases
= 552 total phrases, of which 50% (276) had grammar mistakes.

Following standard text entry study practice, phrases were shown on screen to be copied. We used the phrases defined in section 6.6.

6.3 Participants

Twelve participants (8 female, 4 male) were recruited for the study through advertising to an undergraduate participant pool at SFU. Most were between 18 and 25 years old, with one between 25-35.

In previous studies we observed that native speakers tended to correct minor grammar errors as they were typing in copy tasks and to ignore system feedback if they made mistakes. Thus, we purposefully selected a majority of participants whose first language was not English, with the intent of increasing the probability of copy-writing mistakes. Ten participants had English as a second language and two had it as their first language. The mean IELTS score for English as a second language participants was still fairly high (6.3/9), which can be explained through them being university students. All participants had prior experience with touchscreen-based text entry. Participants completed a consent form before participating in the experiment and were introduced to the general procedure of the study before performing the main tasks.

6.4 Dependent Variables

The study recorded participants' entry speed, accuracy, and the total number of operations per character with *WiseType* and a conventional touchscreen keyboard, which looks and works like most touchscreen device keyboards with the auto-correction feature enabled. As with most touchscreen keyboards, the auto-correction method changes words without providing any noticeable feedback to the user. Only spelling mistakes are displayed to the user using red underlining. Our goal was to compare this type of support in terms of participants' number of operations per character, text entry speed, and error rates with *WiseType's* enhanced visual feedback, grammar correction, and smart correction features. As mentioned above, both conditions were based on the same core system, layout, underlying dictionary and autocorrection, with the only difference being the enhanced visual feedback and smart correction method. We measured performance in terms of speed and error rates.

6.5 Procedure

Before the study, participants were asked to fill out a background questionnaire about their age, gender, English proficiency, and their current touchscreen device keyboard experience, such as, whether text entry using their touchscreen keyboard hinders or improves their speed and writing accuracy compared to a physical keyboard.

During the study, participants were asked to enter 46 phrases using the conventional touchscreen keyboard and *WiseType* conditions (23 phrases per each condition). When done, they were asked to complete a questionnaire, where they could rate and comment on the new keyboard. We correlated this data with observations collected during the experiment as well as insights gathered from a brief post-study interview. Each participant spent between 50 and 60 minutes with the task, depending on their typing speed. We offered breaks in between conditions, but almost all participants chose to continue with the sessions without breaks.

6.6 Phrases and Error Injection

As the pilot study identified problems with the phrased collected from social networks, we created a more controlled set of phrases as stimuli for our text copy task, based on the Enron phrase dataset [48]. The phrases selected were generally short to medium length ranging from 3 to 12 words (M = 6.1, SD = 1.68), containing from 14 to 67 characters (M = 29.9, SD = 10.13). Still, they were easy to remember, a requirement identified by previous work [41].

Though tablets are often used for longer text entry, the use of transcription (which is the de-facto standard in text entry studies) in our experiment requires short, memorable phrases [40]. To the best of our knowledge, the use of longer text entry phrases has only been proposed in two papers, by asking users to compose text based on a scenario prompt [49] or by shortly describing an image presented to the users [35]. Although both studies have shown promise in the use of such evaluation tasks to replace the short phrase transcription task, these methods have not been validated by other work.

We purposely injected grammar errors into 50% of the phrases. We added mistakes in prepositions, verb tense, agreement errors (live/life), singular/plural mistakes, do/does/did issues, and non-infinitive verbs after modals. We shuffled phrases so that participants experienced a mix of unmodified Enron phrases and phrases with injected errors. Examples for injected errors include: "I *does* not know about it", "We are looking for people who *has* expertise in this area", "These people *does* not know the answer", "Today we still are *fly* to space", "We regularly *moving* a conversation from the forum to email" and "I *life* in Moscow".

In line with transcription-based text entry studies, participants were shown phrases one a time and recommended to type them quickly but accurately. They were further instructed to correct any errors they could identify in the phrases. We decided to exclude punctuations in the study, because non-alpha characters introduce a potential confounding source of variation in the dependent measures, and threaten internal validity [32].

7 Results

We used a repeated-measures ANOVA with alpha of 0.05 for all analysis. A Shapiro–Wilk test identified that the assumption of normal distribution was satisfied and all other preconditions of ANOVA were also met.

7.1 Pre-Study Questionnaire Responses

The results of the pre-session questionnaire showed that nine (75%) of all participants had predictive systems, i.e., word prediction and auto-correction, enabled on their either smartphones or tablets. Nine (75%) of all participants responded that their primary mobile operating system was Apple iOS, while three (25%) used Android.

Figure 9 shows the responses of the participants for how they perceived the text entry speed and their typing accuracy with their own, current touchscreen keyboard.

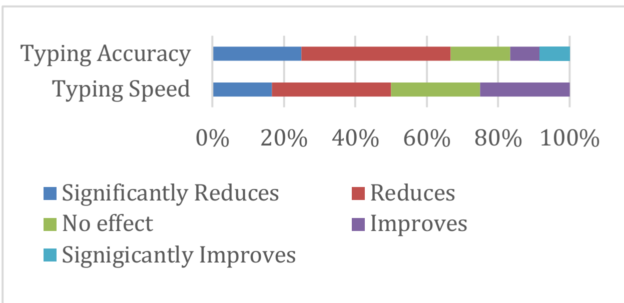

Figure 9: Perception of participants for typing on the touchscreen keyboard of their own device. The majority of participants indicated that their own devices reduce their typing speed and accuracy. Likert scale questions on the y-axis and percentages on the x-axis.

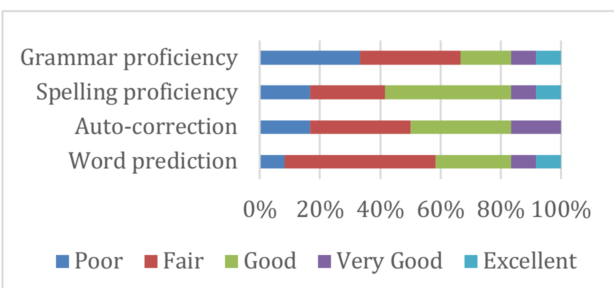

Figure 10: The participants responses on auto-correction, word prediction and their spelling/grammar proficiency when using their own touchscreen keyboard. Likert scale questions on the y-axis and percentages on the x-axis.

We asked participants to rate the word prediction feature of their current mobile keyboard. The majority of participants rated the word prediction as fair, with the remainder spread across the scale. Figure 10 shows the responses of the participants for the questions about auto-correction, word prediction, and spelling/grammar proficiency when using their own touchscreen keyboard.

7.2 Overall Entry Speed

In line with common text entry study protocols, we used words per minute (WPM) metric to measure entry speed [4]. Time was measured from the first keystroke to the last.

There was significant effect on entry speed, $F(1,11) = 10.825$, $p = .007$, with a large effect size (Cohen's $dz = 3.052$) and high power $(1\text{-}\beta) = 1.0$. Mean entry speeds for the conventional and *WiseType* conditions were ($M = 16.76$, $SD = 4.10$) and ($M = 19.87$, $SD = 4.89$), respectively. Figure 11 illustrates the average entry speed for both conditions.

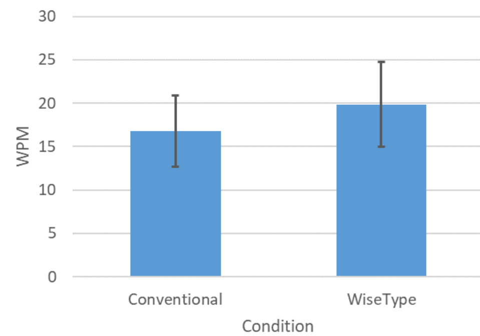

Figure 11: Average entry speed in Words per Minute (WPM) for both conditions.

7.3 Error Rate

To analyze the correctness of the final submitted text, we used the Error Rate (ER) metric [4]. To calculate ER we compared the transcribed phrases with correct ones. While we initially considered using only pre-determined correct phrases, we realized that there were several instances where there were multiple, alternate valid corrections, such as "*Peter did go*" or "*Peter went*". Thus, we checked the transcribed phrases with two independent reliable grammar/spelling checker services (LanguageTool and MS Word) to determine the final ER. For consistency with other work, we ignored all punctuation and capitalization errors.

The mean number of writing errors with the conventional keyboard ($M = 4.52$, $SD = 2.09$) was higher than with *WiseType* ($M = 1.42$, $SD = 1.02$) and the difference was statistically significant, $F(1,11) = 21.27$, $p = .001$, with a large effect size (Cohen's $dz = 1.33$) and high power $(1\text{-}\beta) = 0.996$. Figure 12 illustrates the average error rate for both conditions.

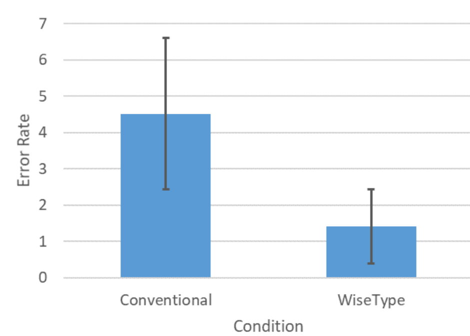

Figure 12: Average Error Rate (ER) for both conditions (y-axis represents percentages).

The final submissions, however, do not show all errors, as users often correct their typing mistakes, and in our case grammar errors, as they type. Corrected errors typically impact typing speed as users need time to correct such errors. To gain a detailed view of how much error correction users performed, we analyzed the Operations per Character (OPC) metric, which is the average number of operation needed to input a single character [2]. This metric is similar to Keystrokes per Character (KSPC), the ratio of the length of the input stream to the length of the transcribed text [4]. The mean of the number of keyboard actions with *WiseType* (M = 1.26, SD = 0.14) was lower than with the conventional keyboard (M = 1.55, SD = 0.25). The difference is statistically significant, $F(1,11) = 16.94$, $p = .002$, with a large effect size (Cohen's dz) = 1.20 and high power $(1-\beta) = 0.987$. Figure 13 illustrates average OPC for both conditions.

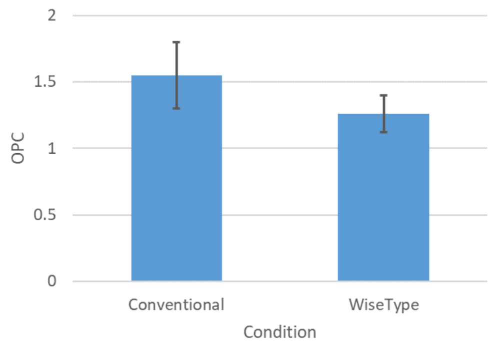

Figure 13: Average Operation per Character (OPC) for both conditions.

Finally, and for measuring error rates, we had observed in previous studies that users sometimes naturally "correct" unusual wordings in standard text collections. Although this experiment was not designed explicitly to compare native and non-native speaker behaviors, we observed that non-native speakers did tend to trust the *WiseType* feedback more than native speakers. In some instances, native speakers saw the orange highlight, which represents a grammar mistake, but did not fix the errors. In post-session interviews, they said they felt that the grammar mistake detection was not accurate, i.e., that there was no mistake even though there were clear grammatical errors. One of the common errors that native speakers overlooked is the use of "*the*", for example in "*Avoid dudes, [the] worst service…*".

7.4 Usage of the Prediction Methods

We observed a significant decrease in the use of regular backspace characters with *WiseType*, $F(1,276) = 41.71$, $p < .001$. The mean number of backspace characters in the conventional and *WiseType* conditions were (M = 7.32, SD = 7.64) and (M = 3.74, SD = 5.15), respectively. Figure 14 shows the average number of backspaces used for both conditions.

The occurrence of auto-correction while using the conventional keyboard ranged from 0 to 7 times per task (M = .60, SD = 1.02), with 78% being correct auto-corrections and 22% incorrect ones. The occurrence of auto-correction while using *WiseType* ranged from 0 to 4 times per task (M = .42, SD = .71), with 81% correct auto-corrections and 19% incorrect ones. This means that we observed that about 1 in 5 auto-corrects were incorrect. The usage of the prediction panel while using the conventional keyboard ranged from 0 to 16 times per task (M = 2.30, SD = 2.97). For *WiseType*, it ranged from 0 to 18 times per task (M = 2.10, SD = 2.77).

For the newer features in *WiseType*, *Smart-Backspace* was used between 0 to 5 times per task (M = 0.32, SD = 0.80). The usage of tapping to access the context menu (for all grammar, spelling, auto-correction and prediction) ranged from 0 to 4 times per task (M = 0.30, SD = 0.55).

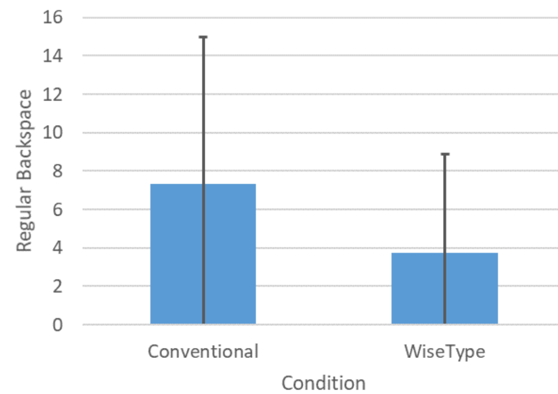

Figure 14: Average number of backspaces used per task for both conditions.

7.5 Post-Study Questionnaire Responses

Figure 15 shows the questionnaire responses and that the majority rated the word prediction feature as very good or good and WiseType overall as very good. Figure 16 illustrates the learnability of *WiseType*.

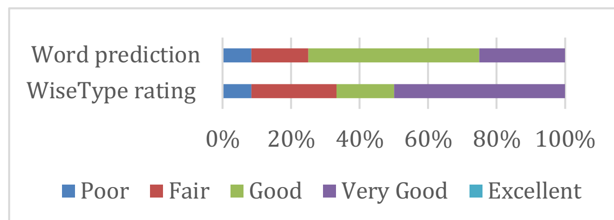

Figure 15 Perception of *WiseType*. The majority of participants rated *WiseType* positively.

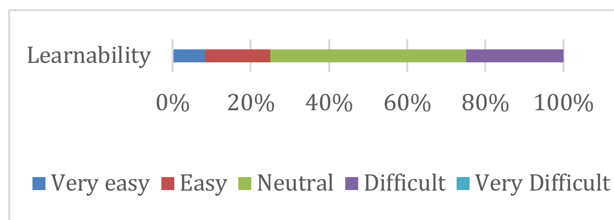

Figure 16 Learnability of *WiseType*. The y-axis represents the used Likert scale and the x-axis percentage.

8 Discussion

The result shows that text entry speed significantly increases (Figure 11) and writing errors decrease significantly (Figure 12) with *WiseType*. Although *WiseType* uses more visible word predictions, which could decrease text entry speed, the availability of fast error correction methods overcomes any potential costs of perception and interaction. We also identified in our pilot studies that each individual feature (highlighting, *Smart-Backspace*, etc.) might not yield a significant improvement. Yet, the main study shows that the synthesis of *Smart-Backspace*, context menu, prediction panel and highlighting yields a significant improvement.

Interestingly, the number of keyboard actions significantly decreases with *WiseType*. One explanation could be that *WiseType* decreases the usage of the regular backspace significantly (Figure 14). Another reason could be that participants depend more on word prediction to complete the words. When we asked participants about auto-correction and word prediction, some commented that they were depending on auto-correction to fix their mistakes. One even stated that "*auto-correction makes people lazy*". We observed that different people might use auto-correction in different ways. In particular, we found that some preferred not to complete words if they knew that the predictive system will complete it.

The pilots and interviews helped us understand participant needs better. Specifically, after identifying in our pilot that participants assumed that underline and highlighted words were interactive and tapped frequently on the touchscreen, we made all highlighted words interactive and added a context menu. We had not considered a context menu before the pilots, especially because tapping and expecting a context menu with predictions is not a very common interaction in current text entry systems on mobile devices.

Performance with our conventional keyboard was below usual typing speeds on mobile phones in the wild [38]. We attribute this to two main reasons: 1) participants were typing on a device and keyboard new to them – whereas they are very familiar with their own device and keyboard in the wild; 2) they were asked to focus on correcting grammatical errors, which requires cognitive effort. This was a non-trivial task for most participants, especially for the non-native English speakers.

In the interviews, we asked participants about their experience with *WiseType* and how it related to other technologies they are using. Some participants compared our keyboard to their mobile devices' keyboard, while others compared it to Grammarly[9]. This last comparison is interesting, as grammar checkers were not widely available on mobile devices at the time of the study. The results for the pre-study questionnaire show that 8 (66.67%) of our participants self-identified their grammar proficiency during text entry as either poor or fair. In the interview, many commented that *WiseType* could help them identify grammar mistakes and typos. Hence, we identify a great potential for integrating grammar checkers into mobile keyboards. The results for the pre-study questionnaire show that 5 (41.6%) rated word prediction in their current touchscreen keyboard positively, while a larger number, 9 (75%), rated *WiseType* in the exit questionnaire positively.

For text entry, the predominant mode of smartphone usage is portrait mode, typically by holding it with one hand and typing with the other [13]. Holding larger tablets (12" and bigger) with a single hand for extended periods is tiring. Fitts' law can be used to predict that text entry performance might be different in smartphones and tablets. This was experimentally validated in a comparison of text entry on smartphones and tablets in landscape mode with a population of older adults (some with tremor) [36]. Still, the benefits of better error visualizations and smart error correction methods are (mostly) independent of the form factor of the device and we expect these benefits to transfer even to smartphone typing in portrait mode. While our results only enable us to make claims on the behaviors associated with our participant pool, previous research has shown that error highlighting mechanisms are beneficial for older adults [26]. That prompts us to reasonably believe that our results will also generalize to other demographics.

9 CONCLUSION

Error correction has been highlighted as a key challenge for text entry as it contributes massively to slow real-life text entry speed and frustration (e.g., [22]). *WiseType* is a new interactive keyboard that offers improved visual feedback to increase error awareness without decreasing text entry speed. The novelty of our approach is the combination of different visual representations for grammar and spelling errors, auto-corrections, and accepted word predictions together with fast correction methods like a context menu and the *Smart-Restorable Backspace*. The fast correction methods decrease the overhead of correcting errors and reduce the number of keyboard actions. Overall, *WiseType* performs better than a conventional predictive keyboard by significantly reducing writing errors and increasing writing speed. Subjective analysis indicates that the highlighting contributed to users actually perceiving and correcting a higher number of grammatical errors far from the current cursor position. Interestingly, our study shows that increasing the visibility of errors combined with fast correction methods can improve both writing speed and accuracy regardless of the potential increase in perception and interaction costs.

Research on error correction is particularly challenging with standard text-entry lab study protocols, as users make very few errors in lab conditions [40,44]. After experimenting with social-network sourced phrases that proved too confusing in pilot studies, our Enron corpus with injected errors proved a useful approach to evaluate error correction methods.

10 FUTURE WORK

We are currently developing a prototype Android IME and are planning to conduct a long-term study to explore users' behaviors and identify additional pattern around error correction. A particular focus will be on how the tradeoff between text entry speed and accuracy is affected by long term usage with our keyboard used as users' standard keyboard. We will also study the types of grammatical error that are best supported and if there are language, regional, cultural, or age variations in perceived importance of different classifications of errors on mobiles.

We also plan to study how our system can support users while learning a language (either as a child or as a second language) and how to tune the support for people with language related learning difficulties, such as dyslexia. We also plan to look further into the language support needs and implications of comparisons of native vs. non-native speaker text entry.

ACKNOWLEDGEMENTS

We are grateful to the participants in our four user studies. The work was partly funded under a scholarship from *King Saud University* to whom we are also grateful.

REFERENCES

[1] Ouais Alsharif, Tom Ouyang, Françoise Beaufays, Shumin Zhai, Thomas M Breuel, and Johan Schalkwyk Google. 2015. Long Short Term Memory Neural Network for Keyboard Gesture Decoding. Icassp: 2076–2080.

[2] Ahmed Sabbir Arif, Sunjun Kim, Wolfgang Stuerzlinger, Geehyuk Lee, and Ali Mazalek. 2016. Evaluation of a smart-restorable backspace technique to facilitate text entry error correction. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems: 5151–5162. https://doi.org/10.1145/2858036.2858407

[3] Ahmed Sabbir Arif, Michel Pahud, Ken Hinckley, and Bill Buxton. 2014. Experimental study of stroke shortcuts for a

[9] https://www.grammarly.com

touchscreen keyboard with gesture-redundant keys removed. In Proceedings of the 2014 Graphics Interface Conference - GI '14, 43–50. https://doi.org/10.20380/GI2014.06

[4] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In Proceedings of the IEEE Toronto International Conference - Science and Technology for Humanity - TIC-STH '09, 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

[5] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2010. Predicting the cost of error correction in character-based text entry technologies. In Proceedings of the 28th international conference on Human factors in computing systems - CHI '10. https://doi.org/10.1145/1753326.1753329

[6] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2013. Pseudo-pressure detection and its use in predictive text entry on touchscreens. In Proceedings of the 25th Australian Computer-Human Interaction Conference on Augmentation, Application, Innovation, Collaboration - OzCHI '13, 383–392. https://doi.org/10.1145/2541016.2541024

[7] Ahmed Sabbir Arif, Cristina Sylla, and Ali Mazalek. 2016. Learning New Words and Spelling with Autocorrections. In Proceedings of the 2016 ACM on Interactive Surfaces and Spaces - ISS '16, 409–414. https://doi.org/10.1145/2992154.2996790

[8] Kenneth C. Arnold, Krzysztof Z. Gajos, and Adam T. Kalai. 2016. On Suggesting Phrases vs. Predicting Words for Mobile Text Composition. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16, 603–608. https://doi.org/10.1145/2984511.2984584

[9] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12, 251. https://doi.org/10.1145/2371574.2371612

[10] Tyler Baldwin and Joyce Chai. 2012. Autonomous Self-Assessment of Autocorrections: Exploring Text Message Dialogues. Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: 710–719. Retrieved from http://aclweb.org/anthology/N12-1089

[11] Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge, and Shumin Zhai. 2012. Bimanual gesture keyboard. In Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12, 137. https://doi.org/10.1145/2380116.2380136

[12] Xiaojun Bi, Barton a. Smith, and Shumin Zhai. 2010. Quasi-qwerty soft keyboard optimization. Proceedings of the 28th international conference on Human factors in computing systems - CHI '10: 283. https://doi.org/10.1145/1753326.1753367

[13] Christina Bröhl, Alexander Mertens, and Martina Ziefle. 2017. How do users interact with mobile devices? An analysis of handheld positions for different technology generations. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-319-58536-9_1

[14] Tao Chen and Min Yen Kan. 2013. Creating a live, public short message service corpus: The NUS SMS corpus. Language Resources and Evaluation 47, 2: 299–335. https://doi.org/10.1007/s10579-012-9197-9

[15] Mark D. Dunlop and Andrew Crossan. 2000. Predictive text entry methods for mobile phones. Personal Technologies: 134–143. https://doi.org/10.1007/BF01324120

[16] Mark Dunlop and John Levine. 2012. Multidimensional Pareto Optimization of Touchscreen Keyboards for Speed, Familiarity and Improved Spell Checking. In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12, 2669–2678. https://doi.org/10.1145/2207676.2208659

[17] Frederic Ehrler, Guy Haller, Evelyne Sarrey, Magali Walesa, Rolf Wipfli, and Christian Lovis. 2015. Assessing the Usability of Six Data Entry Mobile Interfaces for Caregivers: A Randomized Trial. JMIR Human Factors 2, 2: e15. https://doi.org/10.2196/humanfactors.4093

[18] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and its Personalization on Touchscreen Typing Performance. Proceedings of the ACM CHI'15 Conference on Human Factors in Computing Systems 1: 649–658. https://doi.org/10.1145/2702123.2702503

[19] M Goel, a Jansen, T Mandel, S Patel, and J Wobbrock. 2013. ContextType: using hand posture information to improve mobile touch screen text entry. Proceedings of CHI 2013: 2795–2798. https://doi.org/10.1145/2470654.2481386

[20] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, Figure 1: 3817–3821. https://doi.org/10.1145/2858036.2858242

[21] Asela Gunawardana, Tim Paek, and Christopher Meek. 2010. Usability Guided Key-Target Resizing for Soft Keyboards. Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10: 111–118. https://doi.org/10.1145/1719970.1719986

[22] Cl James and Km Reischel. 2001. Text input for mobile devices: comparing model prediction to actual performance. Proceedings of the SIGCHI conference on …: 365–371. https://doi.org/10.1145/365024.365300

[23] Anthony Jameson and Per Ola Kristensson. 2017. Understanding and supporting modality choices. In The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 1. ACM, 201–238. https://doi.org/10.1145/3015783.3015790

[24] Heidi Horstmann Koester and Simon P. Levine. 1994. Modeling the Speed of Text Entry with a Word Prediction Interface. IEEE Transactions on Rehabilitation Engineering 2, 3: 177–187. https://doi.org/10.1109/86.331567

[25] Andreas Komninos, Emma Nicol, and Mark D. Dunlop. 2014. Reflections on design workshops with older adults for touchscreen mobile text entry. Interaction Design and Architecture(s) Journal 20: 70–85.

[26] Andreas Komninos, Emma Nicol, and Mark D. Dunlop. 2015. Designed with older adults to support better error correction in smartphone text entry: the MaxieKeyboard. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct - MobileHCI '15, 797–802. https://doi.org/10.1145/2786567.2793703

[27] Andreas Komninos, Vasilios Stefanis, Athanasios Plessas, and Jeries Besharat. 2013. Capturing urban dynamics with scarce check-in data. IEEE Pervasive Computing 12, 4: 20–28. https://doi.org/10.1109/MPRV.2013.42

[28] Per Ola Kristensson and Keith Vertanen. 2014. The inviscid text entry rate and its application as a grand goal for mobile text entry. In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services - MobileHCI '14, 335–338. https://doi.org/10.1145/2628363.2628405

[29] Sunghyuk Kwon, Donghun Lee, and Min K Chung. 2009. Effect of key size and activation area on the performance of a regional error correction method in a touch-screen QWERTY keyboard. International Journal of Industrial Ergonomics 39, 5: 888–893. https://doi.org/10.1016/j.ergon.2009.02.013

[30] Ju-Hwan Lee and Charles Spence. 2008. Assessing the benefits of multimodal feedback on dual-task performance under demanding conditions. Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1: 185–192. https://doi.org/10.1145/1531514.1531540

[31] Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 8: 707–710.

[32]I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. CHI'03 extended abstracts, 754–755. https://doi.org/10.1145/765968.765971

[33]Jillian Madison. 2011. Damn you, autocorrect!: Awesomely embarrassing text messages you didn't mean to send. Hachette Books, New York City, NY, USA.

[34]Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: a mid-air word-gesture keyboard. Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14: 1073–1082. https://doi.org/10.1145/2556288.2556964

[35]Emma Nicol, Andreas Komninos, and Mark D Dunlop. 2016. A Participatory Design and Formal Study Investigation into Mobile Text Entry for Older Adults. International Journal of Mobile Human Computer Interaction 8, 2: 20–46. https://doi.org/10.4018/IJMHCI.2016040102.oa

[36] Hugo Nicolau and Joaquim Jorge. 2012. Elderly text-entry performance on touchscreens. ASSETS '12 https://doi.org/10.1145/2384916.2384939

[37]Sharon Oviatt, Björn Schuller, Philip R. Cohen, Daniel Sonntag, Gerasimos Potamianos, and Antonio Kruger. 2017. The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 1. https://doi.org/10.1145/3015783

[38] Philip Quinn and Andy Cockburn. 2018. Loss Aversion and Preferences in Interaction. Human–Computer Interaction: 1–48. https://doi.org/10.1080/07370024.2018.1433040

[39] Philip Quinn and Shumin Zhai. 2016. A Cost-Benefit Study of Text Entry Suggestion Interaction. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16: 83–88. https://doi.org/10.1145/2858036.2858305

[40] Shyam Reyal, Shumin Zhai, and Per Ola Kristensson. 2015. Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the Wild. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15: 679–688. https://doi.org/10.1145/2702123.2702597

[41] Sayan Sarcar, Ahmed Sabbir Arif, and Ali Mazalek. 2015. Metrics for Bengali text entry research. In CHI 2015 Workshop on Text Entry on the Edge, 4 pages.

[42] Joseph Sclafani, Timothy F. Tirrell, and Orrin I. Franko. 2013. Mobile tablet use among academic physicians and trainees. Journal of Medical Systems 37, 1. https://doi.org/10.1007/s10916-012-9903-6

[43] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction, 267–280. https://doi.org/10.1007/11555261_24

[44] Amanda L. Smith and Barbara S. Chaparro. 2015. Smartphone Text Input Method Performance, Usability, and Preference with Younger and Older Adults. Human Factors 57, 6: 1015–1028. https://doi.org/10.1177/0018720815575644

[45] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In Proceedings of the conference on Human factors in computing systems - CHI '03, 113–120. https://doi.org/10.1145/642611.642632

[46] Megan Sweeney, Kaavya Paruchuri, and Saul N. Weingart. 2018. Going Mobile: Resident Physicians' Assessment of the Impact of Tablet Computers on Clinical Tasks, Job Satisfaction, and Quality of Care. Applied Clinical Informatics 9, 3: 588–594. https://doi.org/10.1055/s-0038-1667121

[47] Hendrik Uller, Jennifer L Gove, John S Webb, Hendrik Müller, Jennifer L Gove, and John S Webb. 2012. Understanding Tablet Use - A Multi-Method Exploration. Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'12): 1–10. https://doi.org/10.1145/2371574.2371576

[48] Keith Vertanen and Per Ola Kristensson. 2011. A versatile dataset for text entry evaluations based on genuine mobile emails. Proceedings 13th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11: 295. https://doi.org/10.1145/2037373.2037418

[49] Keith Vertanen and Per Ola Kristensson. 2014. Complementing text entry evaluations with a composition task. ACM Transactions on Computer-Human Interaction 21, 2: 1–33. https://doi.org/10.1145/2555691

[50] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry using Sentence-Based Decoding of Touchscreen Keyboard Input. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15: 659–668. https://doi.org/10.1145/2702123.2702135

[51] Shumin Zhai and Per Ola Kristensson. 2008. Interlaced QWERTY: Accommodating ease of visual search and input flexibility in shape writing. In Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08, 593. https://doi.org/10.1145/1357054.1357149

[52] Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: reimagining keyboard interaction. Communications of the ACM 55, 9: 91–101. https://doi.org/10.1145/2330667.2330689